Principales commandes UNIX

1. Format des commandes

Une commande simple se compose :

- d'un **nom**, en général l'abréviation d'un ou plusieurs mots, qui indique la fonction de la commande (ex : pwd pour *p*ath of *w*orking *d*irectory)
- d'**options** (souvent introduites par un tiret), qui sont des informations facultatives transmises à la commande pour particulariser son exécution
- de zéro, un ou plusieurs **arguments** qui indique sur quel(s) objet(s) porte la commande (en général des fichiers)

Une option est souvent indiquée par un seul caractère. Plusieurs options peuvent se mettre à la suite :

Ex :

```
ls –l /etc/passwd
ls –a /etc/passwd
ls –a –l /etc/passwd
ls –al /etc/passwd
```

Par convention, les éléments entre crochets **[ ]** sont facultatifs.
Les options peuvent changer d'un système Unix à l'autre (Linux, BSD, Solaris, etc.). De plus, la plupart des commandes admettent beaucoup d'options. Ainsi, ce document ne présente, pour certaines commandes, que les options les plus courantes qui sont souvent standardisées. Pour plus de détails sur les commandes, se référer au manuel en ligne accessible par la commande **man**, ex : **man passwd** pour obtenir de l'aide sur la commande *passwd*. Certaines commandes disposent d'une option --help qui provoque l'affichage d'un manuel.

2. Commandes utilitaires

passwd	permet de changer de mot de passe.
who	affiche la liste des utilisateurs connectés.
date	affiche la date.
cal <mois> <année>	affiche le calendrier du mois, de l'année spécifiée. Par défaut, années et mois courants.
groups	affiche le(s) groupe(s) auquel appartient un utilisateur.

echo <chaîne>	retourne les arguments donnés. echo est notamment utile pour soumettre des données à un pipe, pour éditer le contenu de variables d'environnement etc ... `echo mathis@dpt-info.u-strasbg.fr > .forward` (crée le fichier de réacheminement des mails) `echo $USER`
finger @<adresse_machine>	Affiche la liste des utilisateurs connectés sur une machine donné.
du	Retourne l'espace disque utilisé dans un répertoire donné (répertoire courant par défaut). L'unité est en demi-blocs (soit 512 caractères), en kilo-octets avec l'option –k
su [nom_login]	changement d'identité, root par défaut
uname	identité du système
man <nom_commande>	aide en ligne
df [fichier_bloc]	affiche les quantités d'espace disques utilisés et disponibles de chaque partition

3. Fichiers

pwd	affiche le nom du répertoire courant.
ls	liste le nom des fichiers. `ls -l` édition du catalogue des fichiers du répertoire courant `ls -la` édition de tous les fichiers du répertoire courant (y compris les fichiers "cachés" commençant par un .) `ls -lt` édition du catalogue par date (à partir de la plus récente) `ls -lR` lecture de la hiérarchie complète des fichiers
cd <répertoire>	permet de se placer dans un répertoire donné. `cd /usr/bin` pour aller dans le répertoire /usr/bin `cd` retour au répertoire d'accueil `cd ~` retour au répertoire d'accueil `cd rep` changement de répertoire relatif `cd ..` se rendre au répertoire père
mkdir <répertoire>	crée un répertoire (Make directory).
rmdir <répertoire>	supprime un répertoire (vide) (Remove directory).
cat [-n] [fich1 ... fichn]	affiche le contenu des fichiers sur la sortie standard. -n pour afficher le numéro des lignes
more <fichier>	édition du fichier au terminal en mode page (<espace> page suivante, <return> ligne suivante, /*chaine* pour rechercher une chaine, b pour remonter une page, q pour quitter)
less <fichier>	équivalent de more avec la possibilité de remonté dans le fichier lu
head [-n] <fichier>	affiche les *n* 1ères lignes du fichier (par défaut n=10)
tail [-n *p*] <fichier>	affiche les *p* dernières lignes du fichier (par défaut p=10). L'option –n +p affiche le fichier privé des p-1 premières lignes. Raccourci tail -100 pour tail –n 100
wc [-l -w –c] <fichier>	retourne le nombre de lignes (l), de mots(w), de caractères(c) dans le fichier indiqué.

cp <fichier1> <fichier2> copie le contenu de <fichier1> dans <fichier2>, en détruisant <fichier2> s'il existait.

On distingue 2 usages :

cp [option] source destination

copie d'un seul fichier, en précisant le chemin et le nom du fichier destination

cp [option] ens-fichiers-source répertoire

copie l'ensemble des fichiers dans le rép. spécifié, en gardant les noms

Principales options :

-R recopie récursive, permet de copier toute une arborescence

-i avertit l'utilisateur de l'existence d'un fichier du même nom et lui demande s'il veut le remplacer.

-p effectue une copie en gardant le propriétaire et le groupe d'origine.

-v affiche en clair le nom des fichiers copiés.

Exemples

`cp -R /home /root/tmp`, crée une copie dans /root/tmp/home

mv <fichier1> <fichier2> renomme ou déplace <fichier1> en <fichier2>

mv <fichier1> <répertoire> Déplace (et éventuellement renomme) <fichier1> dans <répertoire>.

`mv mm.seq ~/data` : déplace le fichier mm.seq dans le répertoire préexistant data

`mv mm.seq ../liens/bar` : renomme mm.seq en bar et le déplace dans le répertoire préexistant liens

touch <fichier> si fichier existe, positionnement de sa date de modification à la date actuelle, sinon crée un fichier vide

rm <fichier> supprime le fichier (Remove).

-i demande de confirmation

-r suppression récursive (si <fichier> est un répertoire)

cmp <fichier1> <fichier2> cmp (compare) édite le numéro des lignes différentes entre deux fichiers.

diff <fichier1> <fichier2> édite les lignes différentes entre deux fichiers.

slocate <chaine> affiche la liste des fichiers contenant *chaine* dans leur chemin absolu.

4. Filtres évolués

cut –c*liste* [fichier]

cut –f*liste* [**-d***c*] [**-s**] [fichier] extrait des parties de lignes dans le fichier, en précisant soit leur position par un rang de caractère (**-c**) ex :-c5-8 (car. de la 5e à la 8e position), soit par un numéro de champ (**-f**). *liste* est une liste d'entiers, séparés par une virgule, et spécifiant les positions précédentes (resp. champs).

Options :

-d*c* *c* est le caractère séparateur des champs (par défaut Tab)

-s supprimer toutes les lignes ne comportant aucun séparateur

ex : cut –d':' –f1 /etc/passwd affiche les utilisateurs référencés dans /etc/passwd

sort [options] [critère] [fichier] tri un fichier.

options :

-b ignore les espaces et tabulations

-d seuls les lettres, chiffres et espaces sont significatifs

-f indifférencie minuscules et majuscules

-n tri selon le critère numérique

-r tri en ordre décroissant

-t*x* le caractère x est le séparateur de champs

-u ne conserve qu'un exemplaire des doublons

critères : +*m*[.*n*] -*p*[.*q*] où *m, n, p et q* sont des entiers par défaut, *m* = *p* = *q* = 0

chaque ligne est découpée en champs (par défaut le séparateur est l'espace ou la tabulation). Le critère s'étend du (*n*+1)ème caractère du (*m*+1)ème champ et se termine au

*q*ème caractère après le dernier caractère du *p*ème champ. Si -*p*[.*q*] est omis, le critère s'étend jusqu'à la fin de ligne.

uniq n'envoie sur la sortie standard qu'un seul exemplaire de plusieurs lignes identiques consécutives.
-c précède chaque ligne du nombre d'occurrences
ex : tail -100 /var/log/httpd-access.log | cut -d' ' -f1 | sort | uniq -c | sort –nr
le fichier /var/log/httpd-access.log contient l'historique des accès au serveur apache, chaque ligne commence par l'@ IP du client. Cette commande affiche les @ IP des clients par ordre d'importance du nombre d'accès.

grep [options] [modèle] [fichier] recherche une chaîne de caractères dans un fichier donné et édite les lignes la contenant. La chaîne trouvée correspond au modèle donnée par une expression régulière.

Options :

-c	écrit uniquement le nombre de lignes contenant l'expression
-f *fichier*	le modèle est extrait du fichier
-i	indifférencie minuscules et majuscules
-l	écrit uniquement le nom des fichiers contenant l'expression
-n	chaque chaîne est précédée de son numéro relatif dans le fichier
-v	écrit toutes les chaînes sauf celles correspondant au modèle

Ex :
`grep GDB *` recherche la chaîne GDB dans tous les fichiers du répertoire courant
`grep cours -i *.txt` recherche la chaîne gdb trouvée indifféremment en minuscules ou majuscules dans tous les fichiers .txt du répertoire courant
`grep error -v *.txt` édite toutes les lignes, exceptées celles contenant le mot error
`grep "valeur de la variable" texte.txt` recherche la chaîne `valeur de la variable` dans le fichier `texte.txt`
`grep 'caractère "A"' table.txt` recherche la chaîne `caractère "A"` dans le fichier table.txt

find chemin … expression recherche de fichier dans une arborescence. Les options permettent de nombreux critères : date de création, de modification, selon l'utilisateur, selon la taille, les droits, etc.

find <répertoire> -name <fichier> -print
forme courante de la commande find
`find . -name mm.seq -print` recherche le fichier mm.seq depuis le répertoire courant
`find ~ -name a.out -print` recherche le fichier a.out depuis le home directory

tr convertit une chaîne de caractères en un autre, la chaîne est lue sur l'entrée standard
ex : `tr "[A-Z]" "[a-z]" < fichier.txt` affiche le contenu de `fichier`.txt en minuscules
`tr -d '\015' < fichier.dos > fichier.unix` supprime les caractères spéciaux 015
`tr –s " " < fichier.txt` réduit les séquences d'espaces à un seul espace
ls –l | tr –s " " | cut –d" " –f6 | sort –nr affiche les dates de modifications des fichiers du répertoire courant de la plus récente à la plus ancienne, sans le tr le numéro du champ de la date est variable.

xargs exécute une commande pour chaque ligne de l'entrée standard
ex : recherche le mot virtual dans tous les .cpp et .h de l'arborescence
find . –regex ".*[cpp, h]$" | xargs grep virtual

5. Droits d'accès aux fichiers

chmod <droits> <fichier> changement des droits d'accès d'un fichier.

chmod <utilisateurs>+/-<droits> <fichier> ou
chmod <droits (octal)> <fichier>
utilisateurs : u (user), g (group), o (other)
droits : r (read), w (write), x (execute)

Ex : `chmod u+r toto` donne le droit de lire le fichier toto à son propriétaire
`chmod g-w toto` supprime le droit d'écriture au groupe
`chmod o+x toto` autorise n'importe qui à exécuter le fichier

Droits d'accès et sa correspondance en octal :

--- : 0	-w- : 2	-wx : 3	rw- : 6
--x : 1	r-- : 4	r-x : 5	rwx : 7

Ex : `chmod 750 *` attribuera : rwxr-x--- à tous les fichiers du répertoire courant

umask <valeur octal> — définit le mode de création de fichiers ou de catalogues. La valeur octal est soustraite (et logique) aux droits par défaut des fichiers (666) et des répertoires (777) à leur création.

6. Processus

ps — affiche les informations sur les processus en cours
`ps -fade` liste complète de tous les processus
signification des champs retournés :

UID	nom de l'utilisateur
PID	numéro d'identification du processus
PPID	numéro d'identification du processus père
STIME	heure à laquelle le processus a été lancé
TIME	temps CPU utilisé par le processus
TTY	terminal de rattachement (dans lequel le proc écrit)
CMD	commande ayant provoqué le lancement du processus
%CPU	en % les ressources du microprocesseur utilisées par le processus
%MEM	en % les ressources en mémoire vive utilisées par le processus.
RSS	mémoire réellement utilisée en ko par le processus.
START	l'heure à laquelle le processus a été lancé.

<ctrl-c> — termine le processus premier-plan courant

<ctrl-z> — stoppe l'exécution du processus premier-plan courant (qui pourra être relancé par bg).

kill -9 <numéro_de_PID> — arrête le processus désigné par son PID

<commande> **&** — lance un programme en arrière plan.
`xterm &` lance une nouvelle fenêtre xterm

jobs — affiche les jobs (et leur numéro) stoppés et/ou passés en arrière-plan.

fg %<numéro_job> — ramène le job courant de l'arrière-plan au premier plan (foreground).

bg %<numéro_job> — relance un job stoppé et le place en arrière-plan (en tache de fond, background).

top — affiche la liste des processus, la charge processeurs(s), l'utilisation mémoire, 'q' pour sortir, <espace> pour réaffichage

7. Compression, archivage

zip/unzip <fichier> — compression/décompression d'un fichier zip

gzip <fichier> compression. Génère un fichier .gz.
gunzip <fichier> décompression d'un fichier .gz.

tar [options] <tarfichier> <répertoire_à_archiver>
archive (tare) un ensemble de fichiers ou les fichiers d'un répertoire en un fichier .tar unique

- f introduit le fichier
- v verbeux, édite les opérations en cours
- c archivage
- t consultation
- x extraction
- z compression

ex : `tar cvf monfich.tar monrep` crée un fichier `monfich`.tar contenant le répertoire monrep
`tar xvf monfich.tar` déploie l'archive
`tar cvfz monfich.tgz monrep` idem que précédemment mais avec compression
tar tvfz monfich.tgz affiche le contenu de l'archive compressée `monfich.tgz`

8. réseau

<host> désigne une adresse IP de machine ou le nom symbolique de celle-ci (ex : pintade.u-strasbg.fr)

nslookup/dig < host > retourne l'adresse IP de la machine (fonctionne en sens inverse avec l'adresse IP en paramètre)
telnet <host> <port> établit la connexion interactive avec un autre ordinateur sur le réseau Internet. Adresse_host correspond au nom de la machine ou son adresse IP.
ssh <[user@]host> idem en mode sécurisé, option –X pour autoriser le fenêtrage
ftp <host> établit la connexion en transfert de fichiers avec un autre ordinateur, help pour obtenir la liste des commandes
scp [[user1@]host1:]fichier1 [[user2@]host2:]fichier2
copie sécurisée à travers le réseau
ifconfig affichage et configuration des cartes réseau
route affichage des routes actives (idem à netstat –r)
ping < host > test de connectivité
traceroute < host > test de transit réseau
netstat affiche des infos réseaux (ports ouverts, etc), (-t ; tcp, -u ;
tcpdump sniffe le trafic sur le réseau, ex :

```
tcpdump src 130.79.90.60 and dst 130.79.44.193 –ne // n : numeric (ip + num port), e : adresses MAC
tcpdump src 130.79.90.60 and port 80  –ne
```

wget aspiration de pages web
nmap <host> affiche la liste des ports ouverts. Ex : nmap localhost (scan localement les ports 1->1024 + ceux précisés dans nmap-services), nmap localhost –p1-65535 (scan des ports 1 à 65535)